AMENDMENTS TO THE CLAIMS

Please amend the pending claims as indicated below:

1. (Currently Amended) A method for interfacing between a multi-threaded application and a restrictive back-end processing system, wherein the back-end processing system requires a thread-dependent connection where a relationship between a connection and an application thread is maintained, the method comprising:

detecting [[a]] the thread-dependent connection in the back-end processing system;

maintaining a single thread to link to the detected thread-dependent connection; and

correlating multiple threads from the multi-threaded application with the maintained single thread, thereby allowing operations requested by the multi-threaded application over the multiple threads to be performed on the back-end processing system through the thread-dependent connection.

2. (Currently Amended) The method of claim 1, wherein detecting a thread-dependent connection comprises detecting the multi-threaded application attempting to access the back-end processing system.

3. (Currently Amended) The method of claim 1, wherein the ~~connector~~ thread-dependent connection allocates memory for connection instances, and wherein detecting a thread-dependent connection comprises detecting the ~~connector~~ thread-dependent connection allocating memory for a connection instance.

4. (Currently Amended) The method of claim 1, wherein detecting a thread-dependent connection comprises reading a header data structure for the ~~connector~~ thread-dependent connection , the header data structure storing identification data including data identifying ~~the~~ thread requirements of the ~~connector~~ thread-dependent connection.

5. (Currently Amended) The method of claim 4, wherein reading a header data structure for the ~~connector~~ thread-dependent connection comprises reading a flag in the header data structure, the

flag indicating whether the ~~connector~~ thread-dependent connection support multiple threads per connection.

6. (Currently Amended) The method of claim 1, wherein maintaining a single thread comprises receiving a thread from the multi-threaded application and isolating the single thread as the single thread to link to the connection.

7. (Currently Amended) The method of claim 1, wherein the thread-dependent ~~connector~~ connection is configured to generate a plurality of simultaneous connections, comprising:

detecting a second thread-dependent connection in the back-end processing system;

maintaining a second single thread to link to the detected second thread-dependent connection; and

correlating one or more second threads from the multi-threaded application with the maintained second single thread, thereby allowing additional operations requested by the multi-threaded application over the second thread(s) to be performed on the back-end processing system through the second thread-dependent connection.

8. (Currently Amended) The method of claim 1, wherein correlating multiple threads from the multi-threaded application comprises mapping each of the multiple threads with the maintained single thread.

9. (Currently Amended) The method of claim 1, wherein correlating multiple threads from the multi-threaded application comprises generating a thread support object to support relationships between the multiple threads and the maintained single thread and using the thread support object to toggle execution of an operation between one of the multiple threads and the maintained single thread.

10. (Original) The method of claim 9, wherein the thread support object comprises two or more semaphores, and comprising using the two or more semaphores to toggle execution of the operation between one of the multiple threads and the maintained single thread.

11. (Currently Amended) A thread consistency support system for providing thread consistency between a multi-threaded application and a thread-dependent connector allocated in a restrictive back-end system, wherein the thread-dependent connector only supports a single thread to link to that connector for operations on that connector, and wherein the multi-threaded application creates multiple threads that attempt to access the connector, the system comprising:

an arbiter layer positioned between the application and the thread-dependent connector, the arbiter layer being configured to receive multiple threads from the application and to produce a single internal thread from the arbiter layer to the thread-dependent connector upon which operations of the multiple threads are performed; and

an activation detector that activates the arbiter layer in response to the activation detector detecting a multi-threaded application attempting to access the restrictive back-end system.

12. (Canceled)

13. (Currently Amended) The system of claim [[12]] 11, wherein the activation detector is an enhancement incorporated into a connector application program interface.

14. (Currently Amended) The system of claim 11, wherein the arbiter layer channels the multiple threads from the application through [[a]] the single internal thread by mapping the threads to preserve one thread per thread-dependent connector.

15. (Currently Amended) The system of claim 11, comprising a thread isolation routine for isolating a thread from the multiple threads of the application to [[a]] the single internal thread for linking to a thread-dependent connection.

16. (Currently Amended) The system of claim 11, comprising a toggle routine channels the multiple threads from the application to produce [[a]] the single internal thread by using threading in connection with to isolate thread execution.

17. (Original) The system of claim 11, wherein the single internal thread produced by the arbiter layer acts as a thread sub-connection to the thread-dependent connector that is assigned to each original multiple thread connection from the application to the arbiter layer.

18. (Original) The system of claim 17, wherein the thread consistency support system utilizes connector methods, and wherein the connector methods appear to the application to be those of the underlying sub-connected thread-dependent connector.

19. (Currently Amended) The system of claim 11, wherein the system establishes a connection handle for the single internal thread with the thread-dependent connector that is returned to [[the]] a connector application program interface of a calling application, and wherein the thread consistency support system utilizes the connection handle to identify and employ the single internal thread when interacting with the thread-dependent connector in response to requests from the multi-threaded application.

20. (Currently Amended) The system of claim 11, wherein the system is configured to channel multiple threads from [[a]] the multi-threaded application to more than one thread-dependent connector.

21. (Original) A thread consistency support system for providing thread consistency from a connector application program interface that creates multiple threads to a thread-dependent connector that only allows a single thread to link to that connector for all operations on that connector, the system comprising:

a threading meta-connector interacting between the connector application program interface and the thread-dependent connector, wherein the threading meta-connector establishes a connection handle for a single internal thread with the thread-dependent connector that is returned to the connector application program interface of a calling multi-threaded application in place of connection handles requested for multiple application threads, in response

to the threading meta-connector's receipt of multiple application threads from the connector application program interface; and

an activation detector that activates the threading meta-connector in response to the activation detector identifying a multi-threaded application attempting to access a thread-dependent connector;

thereby ensuring that the single internal thread that initializes a connection from the thread-dependent connector is used for all subsequent operations attempted by the multiple application threads from the multi-threaded application to that thread-dependent connector.

22. (Currently Amended) A method of providing thread consistency from a connector application program interface that creates multiple threads to a thread-dependent connector that only allows a single thread to link to that connector for all operations on that connector, the method comprising:

receiving multiple application threads from the connector application program interface, the multiple application threads attempting to access the thread-dependent connector;

creating a single internal thread that links with the thread-dependent connector in response to the receipt of the multiple application threads; and

performing data operations of the multiple application threads from the connector application program interface over the single internal thread link with the thread-dependent connector.

23. (New) A computer-readable storing medium, comprising:

a set of instructions, the set of instructions capable of being executed by a processing arrangement to implement a method for interfacing between a multi-threaded application and a restrictive back-end processing system, the back-end processing system requires a thread-dependent connection where a relationship between a connection and an application thread is maintained, the set of instructions effective to perform the steps of:

detecting the thread-dependent connection in the back-end processing system;

maintaining a single thread to link to the detected thread-dependent connection; and

correlating multiple threads from the multi-threaded application with the maintained single thread, thereby allowing operations requested by the multi-threaded application over the multiple threads to be performed on the back-end processing system through the thread-dependent connection.

24. (New) The computer-readable storing medium of claim 23, wherein detecting a thread-dependent connection comprises detecting the multi-threaded application attempting to access the back-end processing system.

25. (New) The computer-readable storing medium of claim 23, wherein the thread-dependent connection allocates memory for connection instances, and wherein detecting a thread-dependent connection comprises detecting the thread-dependent connection allocating memory for a connection instance.

26. (New) The computer-readable storing medium of claim 23, wherein detecting a thread-dependent connection comprises reading a header data structure for the thread-dependent connection, the header data structure storing identification data including data identifying thread requirements of the thread-dependent connection.

27. (New) The computer-readable storing medium of claim 26, wherein reading a header data structure for the thread-dependent connection comprises reading a flag in the header data structure, the flag indicating whether the thread-dependent connection support multiple threads per connection.

28. (New) The computer-readable storing medium of claim 23, wherein maintaining a single thread comprises receiving a thread from the multi-threaded application and isolating the single thread as the single thread to link to the connection.

29. (New) The computer-readable storing medium of claim 23, wherein the thread-dependent connection is configured to generate a plurality of simultaneous connections, comprising:

detecting a second thread-dependent connection in the back-end processing system;

maintaining a second single thread to link to the detected second thread-dependent connection; and

correlating one or more second threads from the multi-threaded application with the maintained second single thread, thereby allowing additional operations requested by the multi-threaded application over the second thread(s) to be performed on the back-end processing system through the second thread-dependent connection.

30. (New) The computer-readable storing medium of claim 23, wherein correlating multiple threads from the multi-threaded application comprises mapping each of the multiple threads with the maintained single thread.

31. (New) The computer-readable storing medium of claim 23, wherein correlating multiple threads from the multi-threaded application comprises generating a thread support object to support relationships between the multiple threads and the maintained single thread and using the thread support object to toggle execution of an operation between one of the multiple threads and the maintained single thread.

32. (New) The computer-readable storing medium of claim 31, wherein the thread support object comprises two or more semaphores, and comprising using the two or more semaphores to toggle execution of the operation between one of the multiple threads and the maintained single thread.

33. (New) A computer-readable storing medium, comprising:

a set of instructions, the set of instructions capable of being executed by a processing arrangement to implement a method of providing thread consistency from a connector application program interface that creates multiple threads to a thread-dependent connector that only allows a single thread to link to that connector for all operations on that connector, the set of instructions effective to perform the steps of:

receiving multiple application threads from the connector application program interface, the multiple application threads attempting to access the thread-dependent connector;

creating a single internal thread that links with the thread-dependent connector in response to the receipt of the multiple application threads; and

performing data operations of the multiple application threads from the connector application program interface over the single internal thread link with the thread-dependent connector.